



日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年 5月11日

出 願 番 号

Application Number:

特願2001-142188

出 願 人

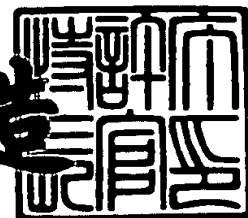
Applicant(s):

三菱電機株式会社

2001年 5月30日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3047579

【書類名】 特許願

【整理番号】 531435JP01

【提出日】 平成13年 5月11日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/38

【発明者】

    【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社  
社内

    【氏名】 原口 喜行

【特許出願人】

    【識別番号】 000006013

    【氏名又は名称】 三菱電機株式会社

【代理人】

    【識別番号】 100089118

    【弁理士】

    【氏名又は名称】 酒井 宏明

【手数料の表示】

    【予納台帳番号】 036711

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

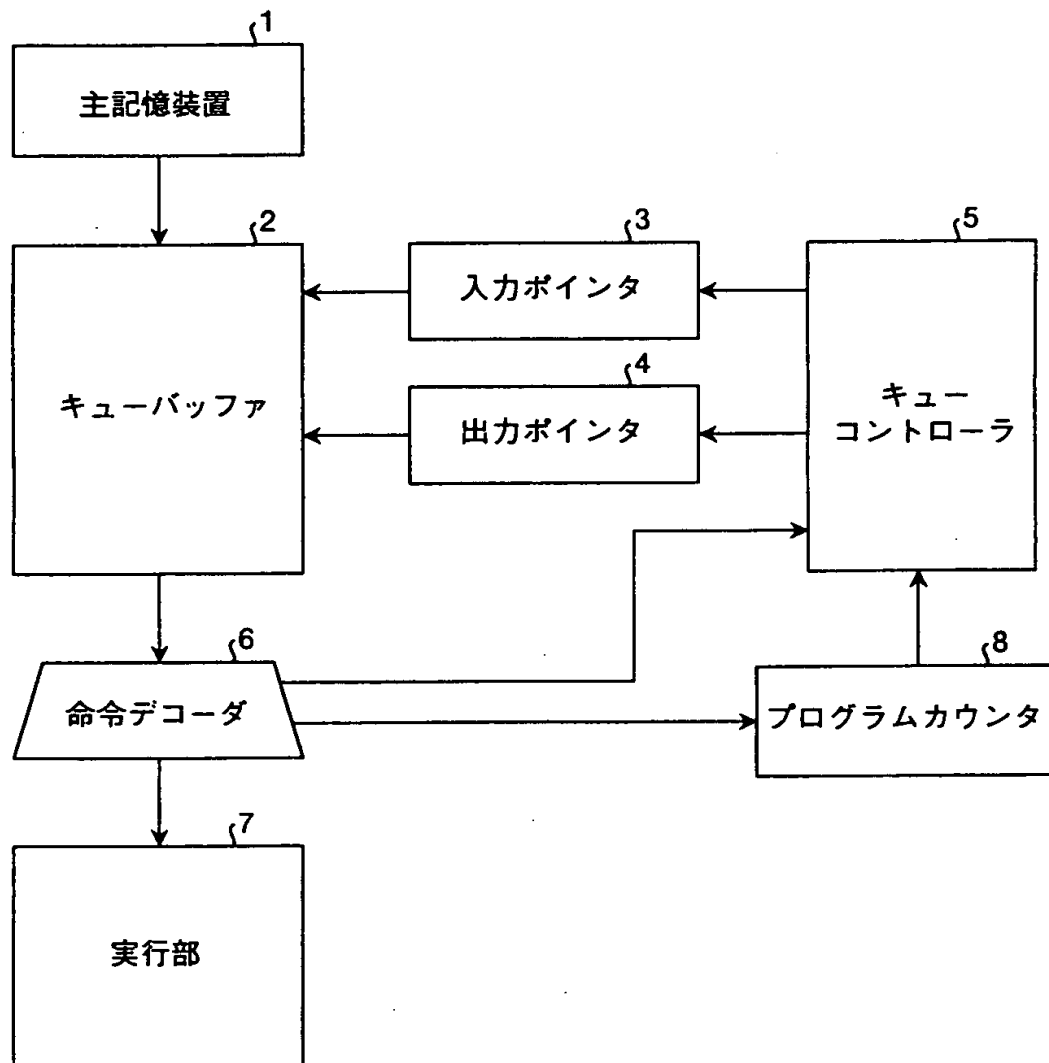
    【物件名】 要約書 1

    【包括委任状番号】 9803092

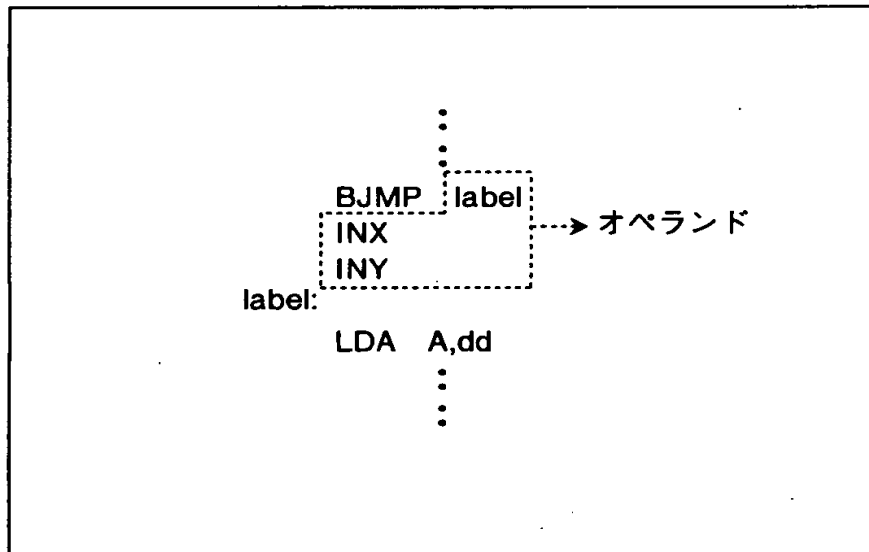
【プルーフの要否】 要

【書類名】 図面

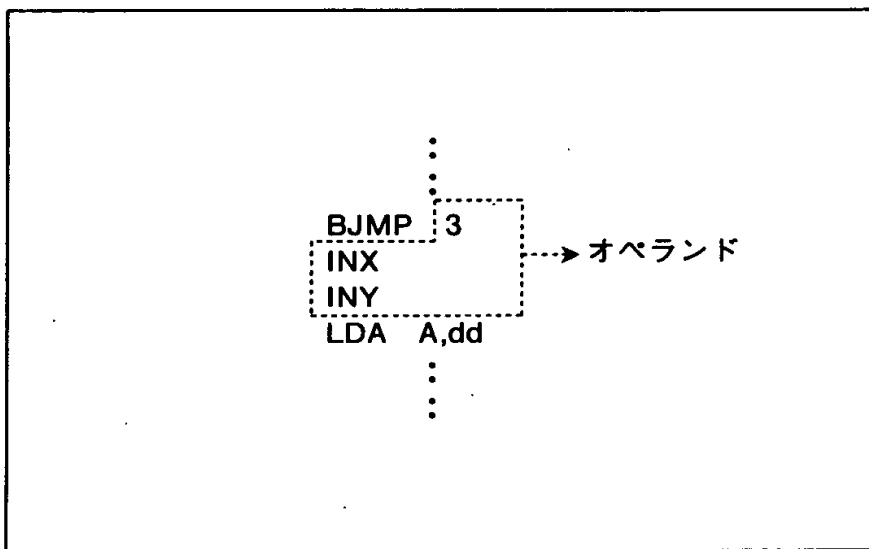
【図 1】



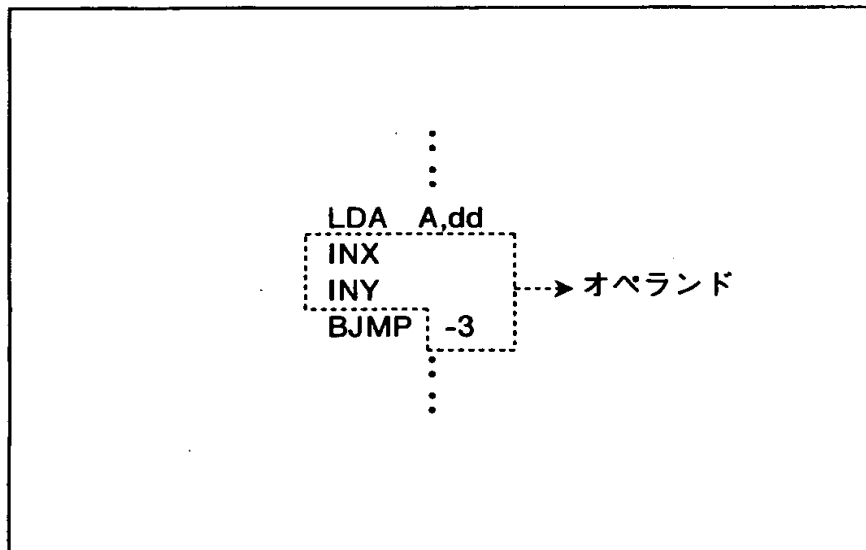
【図 2】



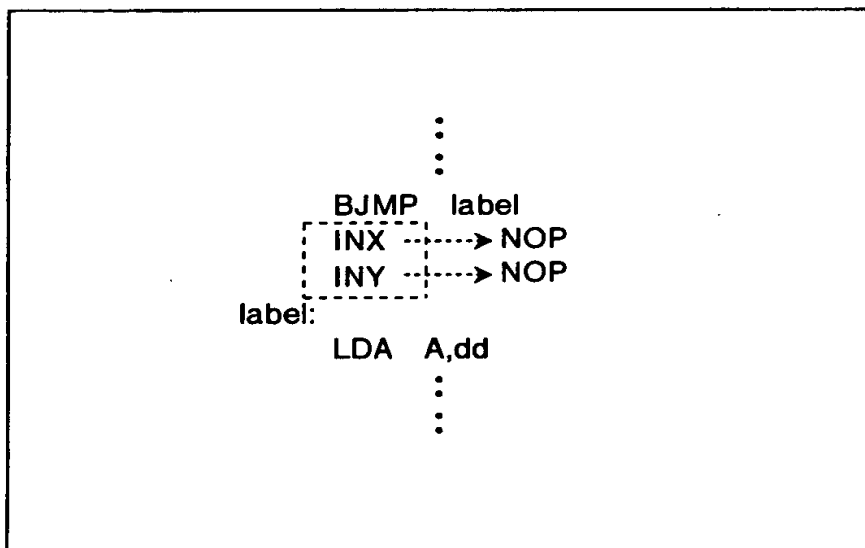
【図 3】



【図 4】



【図 5】



【書類名】 要約書

【要約】

【課題】 簡便な構成で分岐の際の再フェッチを減らすことができ、処理速度を向上させるマイクロプロセッサを得ること。

【解決手段】 新たな分岐命令（B J M P）を追加し、この分岐命令（B J M P）によって指定される分岐先の手前の命令までを当該分岐命令（B J M P）のオペランドとして処理することで処理する命令数を可変とする。前記オペランドとして処理した命令を含む当該分岐命令の命令語長をプログラムカウンタ 8 に出力してプログラムカウンタ 8 のアドレスを更新させるとともに、この分岐命令（B J M P）のときはキューバッファ 2 のフラッシュを行わない。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000006013]

1. 変更年月日	1990年 8月24日
[変更理由]	新規登録
住 所	東京都千代田区丸の内2丁目2番3号
氏 名	三菱電機株式会社

【書類名】 明細書

【発明の名称】 マイクロプロセッサ

【特許請求の範囲】

【請求項 1】 主記憶と、この主記憶からプリフェッチされた命令が格納されるキューバッファと、主記憶上の次に実行する命令が記憶されているアドレスを生成するプログラムカウンタと、前記キューバッファから出力された命令を解読する命令デコーダと、前記プログラムカウンタの出力を用いて前記キューバッファに対する命令の入出力制御を行うキューコントローラ部とを備えたマイクロプロセッサにおいて、

前記命令デコーダは、

前記キューバッファから入力された命令が所定の分岐命令であることを認知すると、この分岐命令によって指定される分岐先の手前の命令までを当該分岐命令のオペランドとして処理し、該オペランド部分を含む当該分岐命令の命令語長を前記プログラムカウンタに出力してプログラムカウンタのアドレスを更新させるとともに前記キューバッファのフラッシュを行わせないようにしたことを特徴とするマイクロプロセッサ。

【請求項 2】 前記分岐先の指定に、ラベルを用いることを特徴とする請求項 1 に記載のマイクロプロセッサ。

【請求項 3】 前記分岐先の指定に、当該分岐命令と分岐先との相対アドレスを指定することを特徴とする請求項 1 に記載のマイクロプロセッサ。

【請求項 4】 前記キューコントローラ部は、現在実行されている命令から予め設定された所定の個数の相対アドレス分だけ前の複数の命令を残すように前記キューバッファの入出力制御を行い、

前記相対アドレスとしてマイナスの相対アドレスを指定可能としたことを特徴とする請求項 3 に記載のマイクロプロセッサ。

【請求項 5】 主記憶と、この主記憶からプリフェッチされた命令が格納されるキューバッファと、主記憶上の次に実行する命令が記憶されているアドレスを生成するプログラムカウンタと、前記キューバッファから出力された命令を解読する命令デコーダと、前記プログラムカウンタの出力を用いて前記キューバッ



ファに対する命令の入出力制御を行うキューコントローラ部とを備えたマイクロプロセッサにおいて、

前記命令デコーダは、

前記キューバッファから入力された命令が所定の分岐命令であることを認知すると、この分岐命令によって指定される分岐先の手前の命令までをNOP命令として処理し、当該分岐命令および前記NOP命令分の命令語長を前記プログラムカウンタに出力してプログラムカウンタのアドレスを更新させるとともに、前記キューバッファのフラッシュを行わせないようにしたことを特徴とするマイクロプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、命令プリフェッチ機能を有するマイクロプロセッサに関し、特にプリフェッチ命令を格納するキューバッファにおける分岐命令実行の際のフラッシュをなくすようにしたマイクロプロセッサに関するものである。

【0002】

【従来の技術】

マイクロプロセッサの高速化の手法として、命令の先取り（プリフェッチ）方式がある。通常の逐次計算機では、連続アドレスに格納されている命令群を順次実行するのが基本である。一方、命令先取り方式では、将来実行が予想される数個先の取り出しを、前の命令の実行や解読と並行して行う。

【0003】

すなわち、あらかじめ主記憶あるいはキャッシュよりプリフェッチされた命令を、高速にアクセス可能な小容量の命令プリフェッチバッファ（キューバッファ）に格納することで、命令フェッチ時のメモリアクセスにより生じる実行の遅れを実質的に削減しようとするものである。

【0004】

従来技術では、分岐命令が実行されると、分岐先アドレスにかかわらず、プリフェッチおよび後続命令の実行を中止し、キューバッファをフラッシュ（クリア

）することにより、そのときまでストアしていた命令を無効にし、分岐先アドレスから新たにプリフェッチしてキューバッファに分岐先命令が格納されるのを待って命令の実行を再開する。

【0005】

このように、従来技術では、分岐命令実行時にキューバッファをフラッシュするようにしているので、プリフェッチ回数が増え、パイプライン処理の乱れ、高速化の妨げとなっていた。

【0006】

そこで、特開平7-73034号公報においては、分岐命令実行時の分岐先アドレスとキューバッファ内に存在する命令の該当アドレス範囲とを比較し、分岐先アドレスが該当アドレス範囲にあるときは、キューバッファをフラッシュすることなくキューバッファ内の命令を用いるようにして、分岐後のプリフェッチ回数を減らすようにしている。

【0007】

【発明が解決しようとする課題】

この従来技術では、確かに分岐後のプリフェッチ回数を減らすことができるが、分岐命令を通常の無条件分岐命令として取り扱っているので、命令デコード後の分岐のためのアドレス生成処理が複雑になり、そのための回路も複雑かつ大規模となる問題がある。

【0008】

この発明は上記に鑑みてなされたもので、内部に複雑な構造を設けることなく無条件分岐の際における不要なキューバッファのフラッシュをなくすようにしたマイクロプロセッサを得ることを目的としている。

【0009】

【課題を解決するための手段】

上記目的を達成するためこの発明にかかるマイクロプロセッサは、主記憶と、この主記憶からプリフェッチされた命令が格納されるキューバッファと、主記憶上の次に実行する命令が記憶されているアドレスを生成するプログラムカウンタと、前記キューバッファから出力された命令を解読する命令デコーダと、前記プ

プログラムカウンタの出力を用いて前記キューバッファに対する命令の入出力制御を行うキューコントローラ部とを備えたマイクロプロセッサにおいて、前記命令デコーダは、前記キューバッファから入力された命令が所定の分岐命令であることを認知すると、この分岐命令によって指定される分岐先の手前の命令までを当該分岐命令のオペランドとして処理し、該オペランド部分を含む当該分岐命令の命令語長を前記プログラムカウンタに出力してプログラムカウンタのアドレスを更新させるとともに前記キューバッファのフラッシュを行わせないようにしたこととを特徴とする。

## 【 0 0 1 0 】

この発明によれば、通常に分岐命令として動作するのではなく、通常データ転送命令、演算命令などと同様に動作する新たな分岐命令（例えば B J M P）がマイクロプロセッサに追加される。この分岐命令（B J M P）によれば、キューバッファ内の命令数が可変となる。すなわち、命令デコーダにおいては、キューバッファから入力された命令が予め設定された所定の分岐命令（B J M P）であることを認知すると、この分岐命令（B J M P）によって指定される分岐先の手前の命令までを当該分岐命令（B J M P）のオペランドとして処理することで、キューバッファ内の命令数を可変とする。そして、該オペランド部分を含む当該分岐命令の命令語長をプログラムカウンタに出力してプログラムカウンタのアドレスを更新させる。また、命令デコーダは、この分岐命令（B J M P）のとき、キューバッファのフラッシュを行わせないようにしている。

## 【 0 0 1 1 】

つぎの発明にかかるマイクロプロセッサは、上記発明において、前記分岐先の指定に、ラベルを用いることを特徴とする。

## 【 0 0 1 2 】

この発明によれば、分岐命令（B J M P）のオペランド部分にラベル（l a b e l）を付加するようにしており、ラベル（l a b e l）で指定された分岐先の手前の命令までが当該分岐命令（B J M P）のオペランドとして処理される。

## 【 0 0 1 3 】

つぎの発明にかかるマイクロプロセッサは、上記発明において、前記分岐先の

指定に、当該分岐命令と分岐先との相対アドレスを指定することを特徴とする。

【 0 0 1 4 】

この発明によれば、分岐命令（B J M P）のオペランド部分に相対番地を付加するようにしており、相対番地で指定された分岐先の手前の命令までが当該分岐命令（B J M P）のオペランドとして処理される。

【 0 0 1 5 】

つぎの発明にかかるマイクロプロセッサは、上記発明において、前記キューコントローラ部は、現在実行されている命令から予め設定された所定の相対アドレス分だけ前の複数の命令を残すように前記キューバッファの入出力制御を行い、前記相対アドレスとしてマイナスの相対アドレスを指定可能としたことを特徴とする。

【 0 0 1 6 】

この発明によれば、キューコントローラ部は、現在実行されている命令から予め設定された所定の個数の相対アドレス分だけ前の複数の命令を残すように前記キューバッファの入出力制御を行っており、これによりキューバッファには、現在実行されている命令より前のアドレスに記憶されている命令が少なくとも前記所定の個数だけは残っていることになる。したがって、分岐命令のオペランド部分にマイナスの相対番地を付加すれば、マイナスアドレス側への分岐が可能となる。

【 0 0 1 7 】

つぎの発明にかかるマイクロプロセッサは、主記憶と、この主記憶からプリフェッチされた命令が格納されるキューバッファと、主記憶上の次に実行する命令が記憶されているアドレスを生成するプログラムカウンタと、前記キューバッファから出力された命令を解読する命令デコーダと、前記プログラムカウンタの出力を用いて前記キューバッファに対する命令の入出力制御を行うキューコントローラ部とを備えたマイクロプロセッサにおいて、前記命令デコーダは、前記キューバッファから入力された命令が所定の分岐命令であることを認知すると、この分岐命令によって指定される分岐先の手前の命令までをNOP命令として処理し、当該分岐命令および前記NOP命令分の命令語長を前記プログラムカウンタに

出力してプログラムカウンタのアドレスを更新させるとともに、前記キューバッファのフラッシュを行わせないようにしたことを特徴とする。

## 【 0 0 1 8 】

この発明によれば、命令デコーダは、キューバッファから入力された命令が予め設定された所定の分岐命令（B J M P）であることを認知すると、この分岐命令（B J M P）によって指定される分岐先の手前の命令までをN O P 命令として処理し、当該分岐命令（B J M P）の命令語長および前記N O P 命令分の命令語長をプログラムカウンタに出力してプログラムカウンタのアドレスを更新させる。また、命令デコーダは、この分岐命令（B J M P）のとき、キューバッファのフラッシュを行わせないようにする。

## 【 0 0 1 9 】

## 【発明の実施の形態】

以下に添付図面を参照して、この発明にかかるマイクロプロセッサの好適な実施の形態を詳細に説明する。

## 【 0 0 2 0 】

図 1 はこの発明にかかるマイクロプロセッサの内部構成例を示すブロック図である。図 1 において、1 は主記憶装置、2 はキューバッファ（命令キュー）、3 は入力ポインタ、4 は出力ポインタ、5 はキューコントローラ、6 はデコーダ、7 は実行部、8 はプログラムカウンタである。キューコントローラ 5、入力ポインタ 3 および出力ポインタ 4 を含めて請求の範囲で言うところのキューコントローラ部を構成している。

## 【 0 0 2 1 】

主記憶装置 1 には、実行するプログラムの命令列が記憶されている。キューバッファ 2 は、例えば 3 2 個の命令を記憶することができるバッファであり、主記憶装置 1 からプリフェッチした命令を格納する。命令デコーダ 6 は、キューバッファ 2 から読み出した命令の命令コードをデコード（解読）する。この命令デコーダ 6 の動作は、本発明の主要をなすので、後で詳述する。

## 【 0 0 2 2 】

実行部 7 には、命令デコーダ 6 の解読結果および命令のオペランド部が入力さ

れる。実行部 7 は、入力されたデコード結果およびオペランド部を用いて、データ転送、四則演算、論理演算、大小比較、シフト演算などの処理を行って命令を実行する。

#### 【 0 0 2 3 】

プログラムカウンタ 8 は、主記憶装置 1 上における次の命令の実行アドレスを格納しているレジスタであり、命令デコーダ 6 から与えられた命令語の長さを現在のプログラムカウンタのカウンタ値（レジスタ値）に加算するよう動作する。

#### 【 0 0 2 4 】

キューコントローラ 5 は、プログラムカウンタ 8 あるいは命令デコーダ 6 の出力に基づいて入力ポインタ 3 および出力ポインタ 4 を制御することで、キューバッファ 2 に対する命令の入出力制御を実行する。入力ポインタ 3 は、キューコントローラ 5 からのライト制御信号によってインクリメント（あるいはデクリメント）されることで、主記憶装置 1 からプリフェッチした命令のキューバッファ 2 に対する書き込み制御を実行する。出力ポインタ 4 は、キューコントローラ 5 からのリード制御信号によってインクリメント（あるいはデクリメント）されることで、キューバッファ 2 に格納された命令の命令デコーダ 6 に対する読み出し制御を実行する。

#### 【 0 0 2 5 】

実施の形態 1 .

つぎに、図 1 の他に図 2 を用いてこの発明の実施の形態 1 について説明する。この実施の形態 1 においては、図 1 のマイクロプロセッサに対し、新たな分岐命令「B JMP」を追加する（図 2 参照）。この分岐命令「B JMP」は、通常分岐命令とは異なる処理を行うが、結果的に通常分岐命令と同様の分岐処理を行うものである。ただし、この分岐命令「B JMP」においては、命令実行の際、キューバッファ 2 はフラッシュされない。図 2 は、キューバッファ 2 に格納された命令列を概念的に示すものである。

#### 【 0 0 2 6 】

この実施の形態 1 においては、分岐命令「B JMP」の分岐先を示すオペランド部にラベル（L a b e l）を指定するようにしている。

## 【0027】

図1の命令デコーダ6では、キューバッファ2から入力された命令を解読して、入力された命令が上記分岐命令「B JMP」であると認識すると、この分岐命令「B JMP」のオペランド部のラベル（l a v e l）で指定される相対番地の手前までに存在する命令（この場合インクリメント命令；I N XおよびI N Y）を分岐命令「B JMP」のオペランドとして処理する。

## 【0028】

すなわち、命令デコーダ6は、キューコントローラ5に対し、出力ポインタ4を更新させるための制御信号を送る。この制御信号によってキューコントローラ5は出力ポインタ4を更新する。この結果、命令デコーダ6には、キューバッファ2から次の命令（この場合I N X）が入力される。この入力された命令を解読して、この命令がラベル（l a b e l）であるか否かを判定する。この場合は、命令（I N X）がラベル（l a b e l）ではないので、命令デコーダ6は、キューコントローラ5に対しさらに制御信号を送って出力ポインタ4を更新させる。このようにして、命令デコーダ6はラベル（L a b e l）を検出するまで、キューコントローラに制御信号を送ってラベル（l a v e l）で指定される相対番地の手前までに存在する命令を読み込む。このようにして、命令デコーダ6は、オペランドとして処理すべき命令部分（この場合I N XおよびI N Y）の命令数およびオペランドとして処理する命令部分の命令語長を検出する。

## 【0029】

そして、命令デコーダ6は、分岐命令「B JMP」の命令語長に、前記オペランドとして処理する命令部分（この場合I N XおよびI N Y）の命令語長を加算し、この加算結果を当該分岐命令「B JMP」の命令語長として、プログラムカウンタ8に出力する。さらに、命令デコーダ6は、キューコントローラ5にフラッシュ禁止のための所定の制御信号を出力する。

## 【0030】

プログラムカウンタ8は、現在のアドレス値に、命令デコーダ6から入力された分岐命令「B JMP」の命令語長を加算して、アドレス値を更新する。また、キューコントローラ5は、命令デコーダ6からの前記制御信号の入力によってキ

キューバッファ2のフラッシュを行わない。

【0031】

したがって、今度は、命令デコーダ6には、キューバッファ2からラベル（label）で指定される相対番地の次の命令（この場合はロード命令；LDA）が読み出されて解読処理されることになり、結果的に通常の分岐命令と同様の処理が行われたことになる。

【0032】

このように、この実施の形態1においては、分岐命令「BJMP」のラベル（label）で指定される分岐先までの命令コードを分岐命令「BJMP」のオペランドとして処理するようにしたので、分岐先までの命令コードが分岐命令「BJMP」の1命令として扱われることになり、結果的に処理される命令数が可変されることになる。すなわち、命令数を可変とすることで無条件分岐と同じ分岐動作が得られることになる。

【0033】

実施の形態1においては、分岐命令「BJMP」は、通常の分岐命令としては動作せず、他のデータ転送命令、演算命令などと同様に動作させかつ通常の分岐命令として扱わずにキューバッファ2のフラッシュを発生させないようにしたので、分岐の際のアドレス演算が従来より簡単になり、これにより内部に複雑な回路構成を設けることなく分岐の際における不要なキューバッファのフラッシュをなくすことができる。したがって、簡便な構成で分岐の際の再フェッチを減らすことができ、処理速度を向上させることができる。

【0034】

なお、図2の場合は、ラベル指定される相対番地までに存在する命令が2つの場合について示したが、この命令数は勿論1または3以上でもよい。また、分岐命令「BJMP」のオペランドの対象となる命令も、図2に示した、インクリメント命令INX, INYに限るわけではない。

【0035】

実施の形態2.

つぎに、図1および図3を用いてこの発明の実施の形態2について説明する。



この実施の形態2においては、図3に示すように、分岐命令「B JMP」の分岐先を相対番地で指定するようにしている。図3においては、分岐命令「B JMP」のオペランド部に相対番地として「3」を指定するようにしている。

## 【0036】

命令デコーダ6では、キューバッファ2から入力された命令を解読して、入力された命令が上記分岐命令「B JMP」であると認識すると、この分岐命令「B JMP」のオペランド部の相対番地（この場合3）で指定される相対番地の手前までに存在する命令（この場合インクリメント命令；INXおよびINY）を分岐命令「B JMP」のオペランドとして処理する。

## 【0037】

すなわち、命令デコーダ6は、キューコントローラ5に対し、分岐命令「B JMP」のオペランド部の相対番地（この場合3）に基づいてオペランドとして処理すべき命令部分（この場合INXおよびINY）の命令数（この場合2）および命令語長を認識する。

## 【0038】

そして、命令デコーダ6は、分岐命令「B JMP」の命令語長に、前記オペランドとして処理した命令部分（この場合INXおよびINY）の命令語長を加算し、この加算結果を当該分岐命令「B JMP」の命令語長として、プログラムカウンタ8に出力する。また、命令デコーダ6は、キューコントローラ5にフラッシュ禁止のための所定の制御信号を出力する。

## 【0039】

プログラムカウンタ8は、現在のアドレス値に、命令デコーダ6から入力された分岐命令「B JMP」の命令語長（INXおよびINYの命令語長を含む）を加算して、アドレス値を更新する。また、キューコントローラ5では、命令デコーダ6からの前記制御信号の入力によってキューバッファ2のフラッシュを行わない。

## 【0040】

したがって、今度は、命令デコーダ6には、分岐命令「B JMP」のオペランド部で指定される相対番地の命令（この場合はロード命令；LDA）がキューバ

ッファ 2 から読み出されて解読処理されることになり、結果的に通常の分岐命令と同様の処理が行われたことになる。

#### 【 0 0 4 1 】

このように、この実施の形態 2 においては、分岐命令「B J M P」のオペランド部に相対番地として指定される分岐先までの命令コードを分岐命令「B J M P」のオペランドとして処理するようにしたので、分岐先までの命令コードが分岐命令「B J M P」の 1 命令として扱われることになり、結果的に処理される命令数が可変されることになる。すなわち、先の実施の形態 1 と同様命令数を可変とすることで無条件分岐と同じ分岐動作が得られるようにしている。

#### 【 0 0 4 2 】

実施の形態 2 においては、分岐命令「B J M P」の結果的なオペランド長を分岐命令「B J M P」の本来のオペランド部で指定された相対番地に対応する長さとする事で通常の分岐命令に相当する動作を行なうようにしたので、分岐の際のアドレス演算が従来より簡単になり、これにより内部に複雑な構成を設けることなく分岐の際における不要なキューバッファのフラッシュをなくすことができる。したがって、簡便な構成で分岐の際の再フェッチを減らすことができ、処理速度を向上させることができる。

#### 【 0 0 4 3 】

なお、図 3 の場合は、ラベル指定される相対番地までに存在する命令が 2 つの場合について示したが、この命令数は勿論 1 または 3 以上でもよい。また、分岐命令「B J M P」のオペランドの対象となる命令も、図 3 に示した、インクリメント命令 I N X, I N Y に限るわけではない。

#### 【 0 0 4 4 】

実施の形態 3.

つぎに、図 1 および図 4 を用いてこの発明の実施の形態 3 について説明する。この実施の形態 3 においては、図 4 に示すように、分岐命令「B J M P」の分岐先を相対番地（この場合は「- 3」）で指定するようにしている。ただし、この場合は、先の実施の形態 2 とは異なり、マイナスアドレス側への分岐を可能としている。

## 【 0 0 4 5 】

この実施の形態 3 では、キューコントローラ 5 は、現在実行されている命令から予め設定された所定の個数（例えば 5 個）の相対アドレス分だけ前の複数の命令を残すように前記キューバッファの入出力制御を行うようにしている。すなわち、入力ポインタ 3 と出力ポインタ 4 に常に少なくとも 5 アドレス分の差をあけて各ポインタ 3, 4 の出力が進むようにしている。

## 【 0 0 4 6 】

命令デコーダ 6 では、キューバッファ 2 から入力された命令を解読して、入力された命令が上記分岐命令「B JMP」であると認識すると、この分岐命令「B JMP」のオペランド部の相対番地（この場合 - 3）で指定される相対番地の手前までに存在する命令（この場合インクリメント命令；I N X および I N Y）を分岐命令「B JMP」のオペランドとして処理する。

## 【 0 0 4 7 】

すなわち、命令デコーダ 6 は、キューコントローラ 5 に対し、分岐命令「B JMP」のオペランド部の相対番地（この場合 - 3）に基づいてオペランドとして処理すべき命令部分（この場合 I N X および I N Y）の命令数および命令語長を認識する。

## 【 0 0 4 8 】

そして、命令デコーダ 6 は、分岐命令「B JMP」の命令語長に、前記オペランドとして処理した命令部分（この場合 I N X および I N Y）の命令語長を加算し、この加算結果にマイナスの符号を付けたものを当該分岐命令「B JMP」の命令語長として、プログラムカウンタ 8 に出力する。また、命令デコーダ 6 は、キューコントローラ 5 にフラッシュ禁止のための所定の制御信号を出力する。

## 【 0 0 4 9 】

プログラムカウンタ 8 は、現在のアドレス値に、命令デコーダ 6 から入力された分岐命令「B JMP」の命令語長を加算して、アドレス値を更新する。また、キューコントローラ 5 では、命令デコーダ 6 からの前記制御信号の入力によってキューバッファ 2 のフラッシュを行わない。

## 【 0 0 5 0 】

したがって、今度は、命令デコーダ 6 には、分岐命令「B J M P」のオペランド部で指定される相対番地（- 3）の命令（この場合はロード命令；L D A）がキューバッファ 2 から読み出されて解読処理されることになり、結果的に通常のマイナスアドレスへの分岐命令と同様の処理が行われたことになる。

## 【 0 0 5 1 】

このように、この実施の形態 3 においても、分岐命令「B J M P」のオペランド部の相対番地で指定されるマイナスアドレスへの分岐先までの命令コードを分岐命令「B J M P」のオペランドとして処理するようにしたので、分岐先までの命令コードが分岐命令「B J M P」の 1 命令として扱われることになり、結果的に処理される命令数が可変されることになる。

## 【 0 0 5 2 】

この実施の形態 3 においては、分岐命令「B J M P」の結果的なオペランド長を分岐命令「B J M P」の本来のオペランド部で指定された相対番地に対応する長さとする事でマイナスアドレスへの分岐命令に相当する動作を行なうようにしたので、分岐の際のアドレス演算が従来より簡単になり、これにより内部に複雑な構成を設けることなく分岐の際における不要なキューバッファのフラッシュをなくすことができる。したがって、簡便な構成で分岐の際の再フェッチを減らすことができ、処理速度を向上させることができる。

## 【 0 0 5 3 】

なお、図 4 の場合は、ラベル指定される相対番地までに存在する命令が 2 つの場合について示したが、この命令数は勿論 1 または 3 以上でもよい。ただし、この上限は、キューバッファ 2 に残す命令の個数によって規定される。また、分岐命令「B J M P」のオペランドの対象となる命令も、図 4 に示した、インクリメント命令 I N X, I N Y に限るわけではない。

## 【 0 0 5 4 】

実施の形態 4.

つぎに、図 1 および図 5 に従ってこの発明の実施の形態 4 について説明する。先の実施の形態 1 ～ 3 では、分岐先までに存在する命令を分岐命令「B J M P」のオペランドとして処理するようにしたが、これらオペランド処理された命令は

、実際には何も命令を実行するわけではないので、ノーオペレーション（NOP）に置き換えても問題はない。

## 【 0 0 5 5 】

例えば、CISC（Complex Instruction Set Computer）のようなアセンブラレベルでの高度な命令セットを提供するプロセッサでは、1命令のアセンブリコードで非常に高度な処理を行なえる。その際、オペランド中にある実際に実行されないただ記述のみがされた命令をNOPに置き換え、実行することができるようにすれば、命令数を変更せずに先の実施形態と同様の分岐動作の実行が可能になる。

## 【 0 0 5 6 】

この実施の形態4においては、図5に示すように、分岐命令「BJMP」の分岐先を示すオペランド部にラベル（Label）を指定するようにしている。

## 【 0 0 5 7 】

この場合、図1の命令デコーダ6では、キューバッファ2から入力された命令を解読して、入力された命令が分岐命令「BJMP」であると認識すると、この分岐命令「BJMP」のオペランド部のラベル（Label）で指定される相対番地の手前までに存在する命令（この場合インクリメント命令；INXおよびINY）をすべてNOPとして処理する。

## 【 0 0 5 8 】

すなわち、命令デコーダ6は、キューコントローラ5に対し、出力ポインタ4を更新させるための制御信号を送る。この制御信号によってキューコントローラ5は出力ポインタ4を更新する。この結果、命令デコーダ6には、キューバッファ2から次の命令（この場合INX）が入力される。この入力された命令INXをNOPとして処理する。さらに、命令デコーダ6は、キューコントローラ5に対し制御信号を送って出力ポインタ4を更新させる。この制御信号によってキューコントローラ5は出力ポインタ4を更新する。この結果、命令デコーダ6には、キューバッファ2から次の命令（この場合INY）が入力される。この入力された命令INYをNOPとして処理する。

## 【 0 0 5 9 】

このようにして、命令デコーダ6はラベル(L a b e l)を検出するまで、キューコントローラ5に制御信号を送ってラベル(l a v e l)で指定される相対番地までに存在する命令を読み込み、これらを全てNOPとして処理する。また、この場合、命令デコーダ6は、キューバッファ2から命令が入力される度に、プログラムカウンタ8に各命令に対応する命令語長を送る。さらに、命令デコーダ6は、キューコントローラ5にフラッシュ禁止のための所定の制御信号を出力する。

## 【0060】

プログラムカウンタ8は、命令デコーダ6から命令語長（この場合は通常の1命令分の語長）が送られてくる度に、この命令語長を現在のアドレス値に加算して、アドレス値を更新する。また、キューコントローラ5は、命令デコーダ6からの前記制御信号の入力によってキューバッファ2のフラッシュを行わない。

## 【0061】

したがって、今度は、命令デコーダ6には、キューバッファ2からラベル(l a v e l)で指定される相対番地の命令（この場合はロード命令；L D A）が読み出されて解読処理されることになり、結果的に命令数を変えない通常の分岐命令と同様の処理が行われたことになる。

## 【0062】

このようにこの実施の形態4においては、分岐命令「B J M P」のラベル(l a b e l)で指定される分岐先までの命令コードをNOPとして処理するようにしたので、命令数を変えることなく無条件分岐と同じ分岐動作が得られるようになる。したがって、分岐の際のアドレス演算が簡単になり、これにより内部に複雑な構成を設けることなく分岐の際における不要なキューバッファのフラッシュをなくすことができる。したがって、簡便な構成で分岐の際の再フェッチを減らすことができ、処理速度を向上させることができる。

## 【0063】

なお、図5の場合は、ラベル指定される相対番地までに存在する命令が2つの場合について示したが、この命令数は勿論1または3以上でもよい。また、NOP処理の対象となる命令も、図5に示した、インクリメント命令I N X, I N Y

に限るわけではない。

【 0 0 6 4 】

ところで、上記の各実施の形態では、命令デコーダ 6 は、分岐命令「B J M P」が入力されたとき、キューコントローラ 5 に対しフラッシュ禁止のための所定の制御信号を出力するようにしているが、特にキューコントローラ 5 に対し制御信号を出力することなくキューバッファ 2 でのフラッシュを禁止させても良い。この場合は、キューコントローラ 5 は、制御信号が入力されたときにフラッシュを行うことになる。

【 0 0 6 5 】

【発明の効果】

以上説明したように、この発明によれば、新たな分岐命令（B J M P）を追加し、この分岐命令（B J M P）によって指定される分岐先の手前の命令までを当該分岐命令（B J M P）のオペランドとして処理することでキューバッファ内の命令数を可変とし、該オペランド部分を含む当該分岐命令の命令語長をプログラムカウンタに出力してプログラムカウンタのアドレスを更新させるとともに、この分岐命令（B J M P）のときはキューバッファのフラッシュを行わせないようにしたので、分岐の際のアドレス演算が簡単になり、これにより内部に複雑な構成を設けることなく分岐の際における不要なキューバッファのフラッシュをなくすことができる。したがって、簡便な構成で分岐の際の再フェッチを減らすことができ、処理速度を向上させることができる。

【 0 0 6 6 】

つぎの発明にかかるマイクロプロセッサによれば、分岐先の指定に、ラベル（L a b e l）を用いるようにしているので、分岐先までの相対番地を指定することなく分岐を行えることができる。また、簡便な構成で分岐の際の再フェッチを減らすことができ、処理速度を向上させることができる。

【 0 0 6 7 】

つぎの発明にかかるマイクロプロセッサによれば、分岐先の指定に、当該分岐命令と分岐先との相対アドレスを指定するようにしており、簡便な構成で分岐の際の再フェッチを減らすことができ、処理速度を向上させることができる。

【 0 0 6 8 】

つぎの発明にかかるマイクロプロセッサによれば、現在実行されている命令から予め設定された所定の相対アドレス分だけ前の複数の命令を残すように前記キューバッファの入出力制御を行うとともに、マイナスの相対アドレスを指定可能としたので、マイナスアドレス側への分岐が可能となる。また、簡便な構成で分岐の際の再フェッチを減らすことができ、処理速度を向上させることができる。

【 0 0 6 9 】

つぎの発明にかかるマイクロプロセッサによれば、新たな分岐命令（B J M P）を追加し、この分岐命令（B J M P）によって指定される分岐先の手前の命令までをN O P命令として処理し、当該分岐命令およびN O P命令の命令語長を前記プログラムカウンタに出力してプログラムカウンタのアドレスを更新させるとともに、この分岐命令（B J M P）のときはキューバッファのフラッシュを行わせないようにしたので、分岐の際のアドレス演算が簡単になり、また分岐の際に命令数が増えることもない。これにより内部に複雑な構成を設けることなく分岐の際における不要なキューバッファのフラッシュをなくすることができる。したがって、簡便な構成で分岐の際の再フェッチを減らすことができ、処理速度を向上させることができる。

【図面の簡単な説明】

【図 1】 この発明にかかるマイクロプロセッサの内部構成例を示すブロック図である。

【図 2】 この発明の実施の形態 1 を説明するための命令列を示す図である。

【図 3】 この発明の実施の形態 2 を説明するための命令列を示す図である。

【図 4】 この発明の実施の形態 3 を説明するための命令列を示す図である。

【図 5】 この発明の実施の形態 4 を説明するための命令列を示す図である。

【符号の説明】



1 主記憶装置、2 キューバッファ（命令キュー）、3 入力ポインタ、4  
出力ポインタ、5 キューコントローラ、6 デコーダ、7 実行部、8 プ  
ログラムカウンタ、B J M P 分岐命令。